



# Modelo Físico de Base de Datos

Centro de Servicios y Gestión Empresarial  
SENA Regional Antioquia

# Conceptualización

# Modelo Físico

El modelo físico es la última fase del diseño de bases de datos, donde se define cómo los datos serán almacenados e implementados en un Sistema de Gestión de Bases de Datos (DBMS) específico como MySQL, PostgreSQL o SQL Server.



# Características

- Específico para un DBMS (se elige MySQL, PostgreSQL, SQL Server, etc.).
- Define tipos de datos para cada columna según el DBMS seleccionado.
- Se establecen índices y claves foráneas para optimizar el rendimiento.
- Incluye restricciones de integridad (UNIQUE, NOT NULL, CHECK).
- Es el modelo listo para implementación en una base de datos real.

# Principales Elementos

# Elementos - Tablas



**Tablas:** Son la estructura básica donde se almacenan los datos en filas (registros) y columnas (atributos). Cada tabla debe tener una Clave Primaria (PK) para identificar de manera única cada fila.

Campos/Columnas				
	Código	Nombre	Apellidos	Identificación
Registro/Fila 1	1	Luis	Cárdenas	123646723
Registro/Fila 2	2	Marta	Sánchez	865446625
Registro/Fila 3	3	Francisco	López	236546625
Registro/Fila 4	4	María	Díaz	89246521

# Elementos - Atributos

**Atributos (Columnas o Campos):** Representan las características de cada entidad. Se les asignan tipos de datos específicos (VARCHAR, INT, DECIMAL, DATE, entre otros).

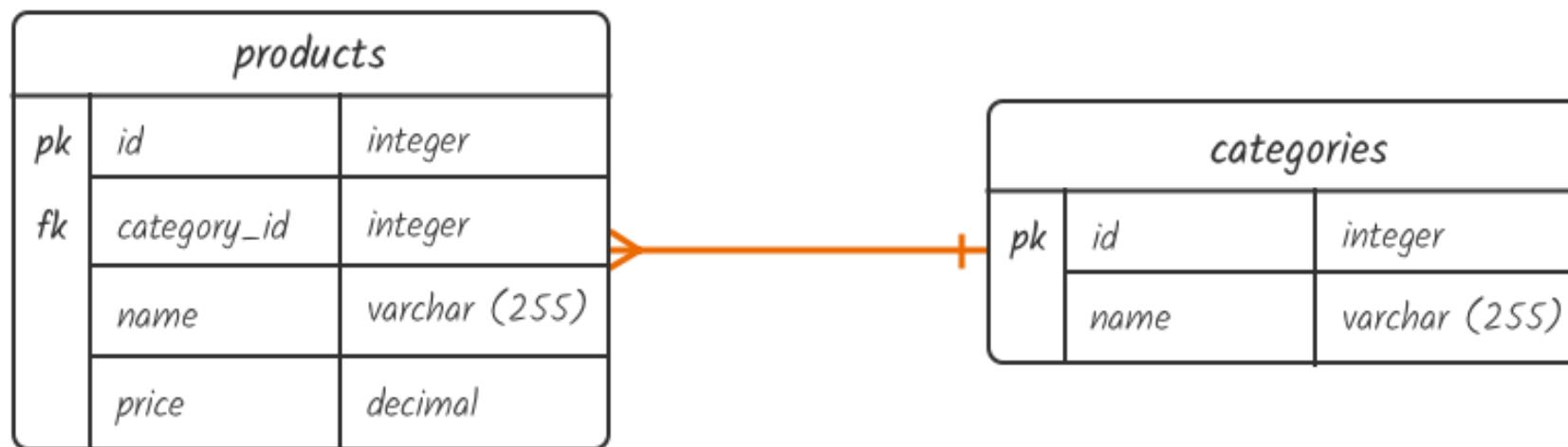
products		
pk	id	integer
fk	category_id	integer
	name	varchar (255)
	price	decimal

<i>id</i>	<i>category_id</i>	<i>name</i>	<i>price</i>
1	1	Pan	300
2	1	Leche	500
3	2	Lavadora	24000

# Elementos - Claves y Restricciones

## Claves:

- **Clave Primaria** (Primary Key - PK): Identifica de manera única cada fila en una tabla.
- **Clave Foránea** (Foreign Key - FK): Relaciona una tabla con otra garantizando la integridad referencial.



# Elementos - Claves y Restricciones



## Restricciones:

- **NOT NULL** → Evita valores nulos.
- **UNIQUE** → Garantiza que los valores sean únicos.
- **CHECK** → Define reglas para los valores.

<i>id</i>	<i>category_id</i>	<i>name</i>	<i>price</i>
1	1	Pan	300
2	1	Leche	500
3	2	Lavadora	24000

# Formas Normales en Bases de Datos Relacionales

# Formas Normales BD

Las formas normales son un conjunto de reglas que se aplican a las bases de datos relacionales para eliminar redundancia, evitar anomalías y mejorar la integridad de los datos.

Se logran mediante un proceso llamado normalización.



# Primera Forma Normal (1FN)

## 1FN - "Eliminar Datos Repetidos":

- Los valores en cada columna deben ser atómicos (no listas o conjuntos); es decir que el número de valores que puede tomar un atributo debe limitarse a uno.
- Cada columna debe tener un solo tipo de dato.
- Debe existir una clave primaria (PK).

ID_Cliente	Nombre	Teléfono	Correo
1	Juan	3001234567, 3115678923	juan@email.com

# Primera Forma Normal (1FN)

**1FN - "Eliminar Datos Repetidos"**

**✗ Ejemplo Incorrecto (No cumple 1FN)**

ID_Cliente	Nombre	Teléfono	Correo
1	Juan	3001234567, 3115678923	juan@email.com

**✓ Ejemplo correcto (Cumple 1FN)**

ID_Cliente	Nombre	Teléfono	Correo
1	Juan	3001234567	juan@email.com
1	Juan	3115678923	juan@email.com

# Segunda Forma Normal (2FN)

## **2FN - "Eliminar Dependencias Parciales":**

- Todos los atributos deben depender completamente de la clave primaria (PK).
- Se eliminan dependencias parciales dividiendo la tabla en varias relacionadas

## **✗ Ejemplo Incorrecto (No cumple 2FN)**

ID_Pedido	ID_Cliente	Nombre_Cliente	Producto	Cantidad
1	101	Juan	Laptop	1
2	102	Ana	Celular	2

# Segunda Forma Normal (2FN)

## 2FN - Eliminar Dependencias Parciales"

### Ejemplo correcto (Cumple 2FN)

ID_Cliente	Nombre_Cliente
101	Juan
102	Ana

**X Problema:** El Nombre\_Cliente depende solo de ID\_Cliente, no de ID\_Pedido.

ID_Pedido	ID_Cliente	Producto	Cantidad
1	101	Laptop	1
2	102	Celular	2

# Tercera Forma Normal (3FN)

## **3FN - "Eliminar Dependencias Transitivas":**

- Debe cumplir 2FN.
- Los atributos no clave deben depender solo de la clave primaria.
- Se eliminan dependencias transitivas moviendo los atributos a nuevas tablas.

## **✗ Ejemplo Incorrecto (No cumple 3FN)**

ID_Pedido	ID_Cliente	Ciudad	Código_Postal
1	101	Bogotá	110111
2	102	Medellín	050022

# Tercera Forma Normal (3FN)

**3FN - "Eliminar Dependencias Transitivas":**

**Ejemplo Correcto (Cumple 3FN)**

ID_Cliente	Ciudad	Código_Postal
101	Bogotá	110111
102	Medellín	050022

Problema: Ciudad y Código\_Postal dependen de ID\_Cliente, pero Código\_Postal depende de Ciudad (relación transitiva).

ID_Pedido	ID_Cliente
1	101
2	102

# Forma Normal de Boyce-Codd (BCNF)



## **Forma Normal de Boyce-Codd (BCNF) - "Estricta 3FN":**

- Debe cumplir 3FN.
- Si hay dependencias funcionales que no provienen de la clave primaria, se reorganizan las tablas.

### **✗ Ejemplo Incorrecto (BCNF)**

ID_Profesor	Materia	Aula
1	Álgebra	101
1	Cálculo	102
2	Física	103

# Forma Normal de Boyce-Codd (BCNF)



**Forma Normal de Boyce-Codd (BCNF) - "Estricta 3FN":**

## ✗ Ejemplo Incorrecto (BCNF)

ID_Profesor	Nombre_Profesor
1	Carlos
2	Laura

✗ **Problema:** Un profesor puede dar varias materias, pero el aula depende de la materia, no del profesor.

Materia	Aula	ID_Profesor
Álgebra	101	1
Cálculo	102	1
Física	103	2

# Tercera Forma Normal (3FN – N:M)

**Ejemplo de 3FN donde se Rompe una Relación de Muchos a Muchos (M:N)**

**✗ Ejemplo Incorrecto (3FN – N:M)**

ID_Estudiante	Nombre_Estudiante	ID_Curso	Nombre_Curso	Profesor
1	Juan	101	Matemáticas	Prof. López
2	Ana	101	Matemáticas	Prof. López
3	Pedro	102	Historia	Prof. García
4	María	102	Historia	Prof. García

# Tercera Forma Normal (3FN – N:M)

**¿Por qué este diseño NO está en 3FN?**

- **Redundancia de datos** → Se repiten los nombres de los cursos y profesores en cada fila.
- **Dependencia Transitiva** → El campo Profesor depende de Curso, no del estudiante.
- **Relación Muchos a Muchos (M:N)** → Un estudiante puede tomar varios cursos, y un curso puede tener varios estudiantes.

# Tercera Forma Normal (3FN – N:M)

## Ejemplo Correcto (Cumple 3FN – N:M)

ID_Estudiante	Nombre_Estudiante
1	Juan
2	Ana
3	Pedro
4	María

ID_Curso	Nombre_Curso	ID_Profesor
101	Matemáticas	201
102	Historia	202

ID_Profesor	Nombre_Profesor
201	Prof. López
202	Prof. García

ID_Estudiante	ID_Curso
1	101
2	101
3	102
4	102

# Tener en cuenta

- **1FN** → Elimina datos repetidos en celdas.
- **2FN** → Elimina dependencias parciales (atributos deben depender completamente de la clave primaria).
- **3FN** → Elimina dependencias transitivas (atributos dependen solo de la clave primaria).
- **BCNF** → Mejora aún más la coherencia y organización.
- La **normalización** ayuda a mantener bases de datos más eficientes, organizadas y sin redundancias.

# Cuarta Forma Normal (4FN)

## 4FN - Eliminación de Dependencias Multivaluadas

- Debe cumplir con 3FN.
- Elimina dependencias multivaluadas (cuando un atributo depende de una clave primaria pero es independiente de otros atributos).
- Una tabla no debe contener conjuntos de valores repetitivos para una misma clave primaria.

# Cuarta Forma Normal (4FN)

## 4FN - Eliminación de Dependencias Multivaluadas

### X Ejemplo Incorrecto (4FN)

- Un estudiante puede estar inscrito en varios cursos y deportes al mismo tiempo, pero curso y deporte no están relacionados entre sí.
- La dependencia entre ID\_Estudiante y Curso es independiente de la relación entre ID\_Estudiante y Deporte.

ID_Estudiante	Curso	Deporte
1	Matemáticas	Fútbol
1	Matemáticas	Baloncesto
1	Historia	Fútbol
2	Química	Natación

# Cuarta Forma Normal (4FN)

## 4FN - Eliminación de Dependencias Multivaluadas

### Ejemplo correcto (4FN)

ID_Estudiante	Curso
1	Matemáticas
1	Historia
2	Química

✓ Beneficio: Ahora, cada relación tiene una única dependencia con la clave primaria, evitando repeticiones innecesarias.

ID_Estudiante	Deporte
1	Fútbol
1	Baloncesto
2	Natación

# Quinta Forma Normal (5FN)

## 5FN - Eliminación de Dependencias de Unión

- Debe cumplir con 4FN.
- Elimina redundancias indirectas cuando una tabla depende de combinaciones de claves de otras tablas.
- Se usa cuando una tabla grande puede descomponerse en varias tablas más pequeñas sin perder información.

### X Ejemplo Incorrecto (5FN)

Problema: Aquí tenemos una relación M:N:N entre Proveedor, Producto y Distribuidor. Hay redundancia porque los proveedores y productos pueden asociarse de diferentes maneras con los distribuidores.

ID_Proveedor	ID_Producto	ID_Distribuidor
1	A	X
1	B	X
2	A	Y
2	B	Y

# Quinta Forma Normal (5FN)

## 5FN - Eliminación de Dependencias de Unión

### Ejemplo Correcto (5FN)

ID_Proveedor	ID_Producto
1	A
1	B
2	A
2	B

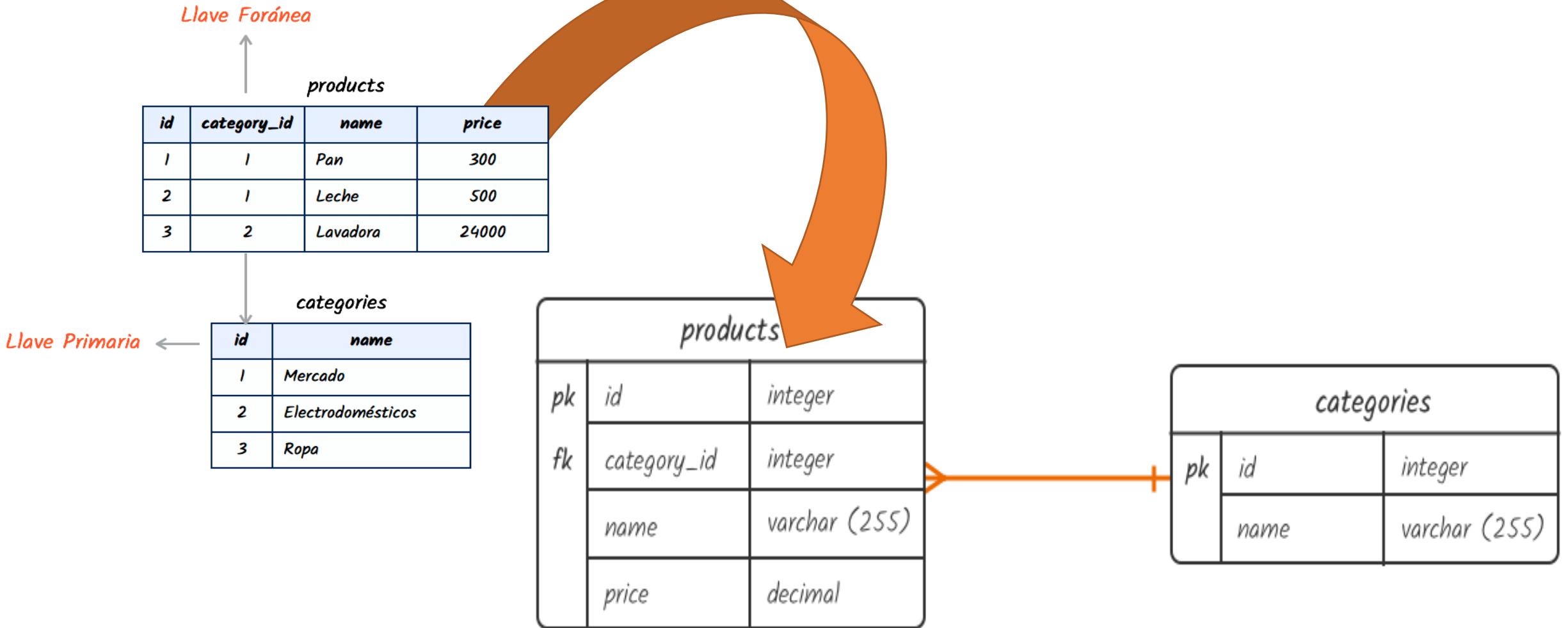
ID_Producto	ID_Distribuidor
A	X
B	X
A	Y
B	Y

ID_Proveedor	ID_Distribuidor
1	X
2	Y

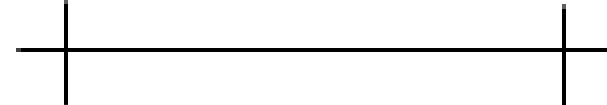
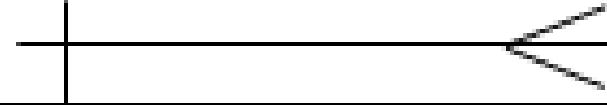
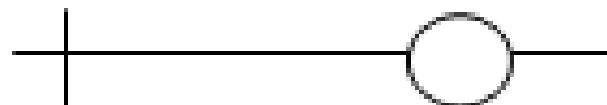
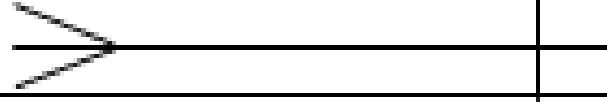
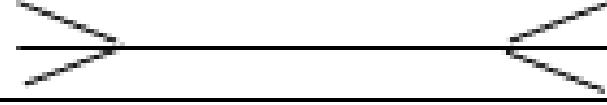
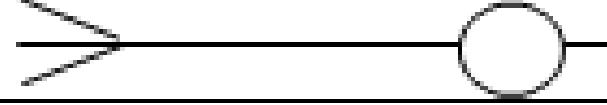
# Formas Normales

Forma Normal	Reglas	Propósito
1FN	No permite datos repetidos en una celda.	Asegurar que cada atributo contenga valores atómicos.
2FN	Elimina dependencias parciales.	Todos los atributos deben depender completamente de la clave primaria.
3FN	Elimina dependencias transitivas.	Los atributos deben depender solo de la clave primaria, no de otros atributos.
4FN	Elimina dependencias multivaluadas.	Evita que una tabla almacene atributos independientes en un solo registro.
5FN	Elimina dependencias de unión.	Descompone tablas complejas en múltiples relaciones sin redundancia.

# Modelo Físico



# Modelo Físico - Elementos

Cardinalidad	Se lee	Representación
1:1	Uno a uno	
1:M	Uno a muchos	
1:0	Uno a ninguno	
M:1	Muchos a uno	
M:M	Muchos a muchos	
M:0	Muchos a ninguno	

# Modelo Físico - Elementos

## Tipos de Datos Numéricos

Se utilizan para almacenar **valores numéricos**, ya sean enteros o decimales.

Tipo	Descripción	Ejemplo de Valor
INTEGER (INT)	Números enteros (positivos o negativos).	100, -50
SMALLINT	Entero pequeño, ocupa menos espacio que INT.	32000
BIGINT	Enteros grandes (útil para ID únicos).	9223372036854775807
DECIMAL (NUMERIC)	Números decimales con precisión exacta.	123.45
FLOAT (REAL, DOUBLE PRECISION)	Números de punto flotante (aproximación).	3.14159

# Modelo Físico - Elementos

## Tipos de Datos de Texto o Cadenas

Se usan para almacenar **palabras, nombres, descripciones, entre otros.**

Tipo	Descripción	Ejemplo de Valor
CHAR(n)	Cadena de longitud fija n.	'ABC' (si es CHAR(5), se almacena 'ABC ')
VARCHAR(n)	Cadena de longitud variable (hasta n).	'Hola Mundo'
TEXT (CLOB, LONGTEXT)	Almacena textos largos (artículos, descripciones).	'Este es un texto largo...'

# Modelo Físico - Elementos

## Tipos de Datos de Fecha y Hora

Se usan para **registrar fechas y tiempos**.

Tipo	Descripción	Ejemplo de Valor
<b>DATE</b>	Solo fecha (Año-Mes-Día).	'2024-03-03'
<b>TIME</b>	Solo hora (Horas:Minutos:Segundos).	'14:30:00'
<b>DATETIME</b>	Fecha y hora combinadas.	'2024-03-03 14:30:00'
<b>TIMESTAMP</b>	Fecha y hora con zona horaria y soporte para marcas de tiempo.	'2024-03-03 14:30:00 UTC'

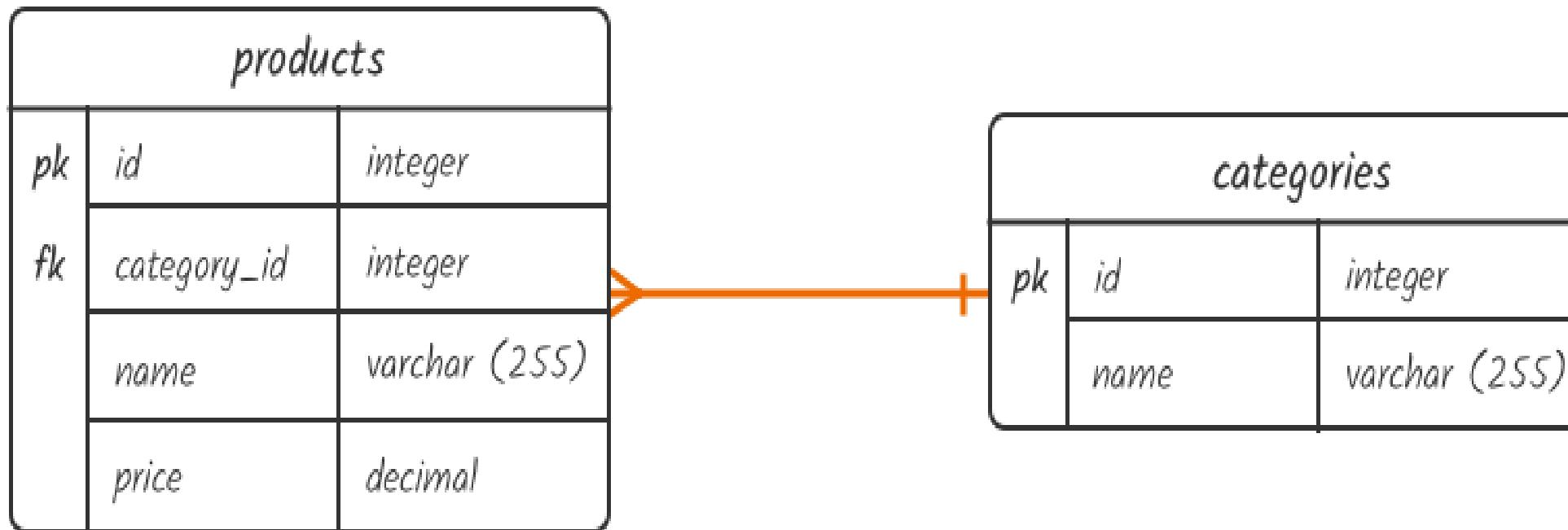
# Modelo Físico - Elementos

## Tipos de Datos Booleanos

Se usan para representar **valores de verdadero/falso**.

Tipo	Descripción	Ejemplo de Valor
BOOLEAN	Valor lógico (true/false).	TRUE, FALSE
BIT (1)	Similar a BOOLEAN, almacena 0 o 1.	0, 1

# Modelo Físico





# G R A C I A S

Presentó: Alvaro Pérez Niño

Instructor Técnico

Correo: aperezn@sena.edu.co

<http://centrodesserviciosygestionempresarial.blogspot.com/>

Línea de atención al ciudadano: 01 8000 910270

Línea de atención al empresario: 01 8000 910682



[www.sena.edu.co](http://www.sena.edu.co)